DOCUMENT RESUME

ED 423 845 IR 019 066

AUTHOR Krawchuk, Cheryl
TITLE Meeting Software Needs of Public and Higher Education: A Collaborative Model.
PUB DATE 1998-02-00
NOTE 8p.; In: Proceedings of Selected Research and Development Presentations at the National Convention of the Association for Educational Communications and Technology (AECT) Sponsored by the Research and Theory Division (20th, St. Louis, MO, February 18-22, 1998); see IR 019 040.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *College School Cooperation; College Seniors; Communication (Thought Transfer); Computer Assisted Instruction; Computer Science Education; *Computer Software Development; *Cooperative Programs; *Courseware; Educational Technology; Elementary Secondary Education; Higher Education; *Instructional Design; Models; Public School Teachers; Student Needs; *Student Projects
IDENTIFIERS Prototypes; Teacher Needs

ABSTRACT

This paper discusses the ongoing prototype of the Software Collaboration Model designed to meet the needs of both public and higher education. The model is based on instructional design principles and proposes that both public and higher education institutions can benefit through the collaboration of senior software design students and inservice public school teachers. Topics discussed include the rationale behind the project, participants, the precautionary measures taken to prevent either collaborating party's needs from being ignored, and the phases of the Software Collaborative Model (interest, equipment, program design, development, testing/training, and implementation). Findings from the initial prototype indicate that the model needs to be modified, giving more attention to communication between all involved parties, instructional design, and allowing for revisions of each step of the process; such modifications have been made, and a revised model is suggested. Two figures illustrating the initial prototype and the revised model are included. (DLS)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 423 845

Meeting Software Needs of Public and Higher Education: A Collaborative Model

By:

Cheryl Krawchuk
University of Arkansas

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

BEST COPY AVAILABLE

PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
M. Simonson

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC).

IR019066

Meeting Software Needs of Public and Higher Education: A Collaborative Model

Cheryl Krawchuk
University of Arkansas

Abstract

While collaboration between public schools and higher education is prevalent due to the Professional Development movements, one area of potential collaboration has been overlooked. This area consists of the need in public schools for educational software and the need in computer education for senior software design projects. This paper discusses the ongoing prototype of the Software Collaboration Model designed to meet the needs of both public and higher education. The model is based on instructional design principles and proposes that both public and higher education institutions can benefit through the collaboration of senior software design students and in-service public school teachers. Findings from the initial prototype indicate that the model needs to be modified, giving more attention to communication between all involved parties, instructional design, and allowing for revisions of each step of the process. Such modifications have been made and a revised model is suggested.

Rationale

The idea to develop a model for collaboration between in-service teachers and senior computer science majors emanated out of separate informal conversations with both parties. Although the conversations varied in perspective, the underlying theme (the need for software development) was the same. The in-service teachers wanted software developed specifically for their students' needs while the computer science students needed to develop a high quality software program. Given the common interest in software development and the complimentary nature of their needs, the possibility of collaboration arose.

To verify the potential for a collaborative model, it was necessary to investigate the needs of both parties and identify the benefits of a collaboration. Hence, an informal needs assessment consisting of individual and group interviews and a benefits analysis was performed. Both in-service teachers and senior computer science design students were assessed prior to the creation of this collaborative model.

In assessing the technology needs for public school teachers, in-service teachers from a rural middle school were the participants. Subsequent to the assessment, one obvious area of need that emerged was for relevant and affordable educational software. Most teachers indicated a lack of funding for software, software that is relevant to specific curricular needs, or both. This finding supports previous literature that suggest teachers and schools lack content-suitable software and/or funding for such software (Braswell, 1994; Oughton & Krawchuk, 1997).

In conjunction with the need for content-suitable software came the perceived need to improve math and science scores across the state. The teachers indicated that standard test scores were below acceptable national and state levels. These feelings were validated recently as the state SAT-9 test reported that over 60% of graduating seniors did not pass the math portion of the test. As such, during the interviews math was identified by the teachers as a content area in need of software assistance.

With the lack of funding and available software, teachers identified a third area of need. They reported that, although they are willing to use supplemental software in their classrooms, they are not capable of creating appropriate software due to time constraints and/or a lack of skills. While these concerns are cogent, the teachers' references to time and skills are not new and have been identified in previous research as areas of consideration (Golas, 1993; Maddux, 1994; Oughton & Krawchuk, 1997). In summary, the findings of the needs assessment on in-service teachers at a rural middle school revealed that these in-service teachers want math-related educational software that is inexpensive and content -relevant without having to create the time or skills necessary to develop it themselves.

The second stage of the needs assessment focused on senior software design students at a mid-sized public university. When assessing the senior software design students, two areas of need were identified. First, students were required to create an extensive software program as a final project in order to graduate. The students viewed this area as "THE" need. They felt as though all other needs were secondary due to the fact that their careers were determined by this one project. Thus, their primary focus was to complete a high-quality final project in a manner that allowed for their graduation.

IR019066

Secondly, because of the high quality of these software projects, several students were interested in marketing their products. However, they felt they would encounter difficulty when trying to find subjects to Beta test their software products. They feared that they would be unable to find an adequate number of subjects as well as subjects that were appropriate for the content. To summarize, the needs of the software design students included the production of a high quality software program and the Beta testing of that program by adequate and appropriate subjects before marketing.

After examining the stated needs of these two parties, the potential benefits of a collaboration became apparent. The teachers could benefit from the collaboration by obtaining free and relevant math-based software programs for their students. At the same time, the software design students could benefit by meeting their degree requirements (producing a high quality program) and obtaining a subject pool for Beta testing of their program. Given the potential win-win outcome from a collaboration, the development of a collaborative model began.

Participants

Upon completion of the needs assessment, a team of software design students and an in-service teacher were chosen to work together on a software development project. The software design team consisted of three senior software design students. They were required to collaboratively create a high quality software program as a final project before graduation. They were also very interested in marketing their final product and wanted to focus on an educational program due to a large market for educational software programs. Both of these factors acted as strong motivators for the students to participate in this project.

The decreasing math scores and need for educational software motivated the in-service teacher to participate. She teaches 5-8 grade middle school mathematics students on a daily basis. She appeared extremely interested in attaining free and relevant software for her students and consistently indicated her need for computer-assisted instruction for her students. As such, she immediately became actively involved in this project.

Software Collaboration Model

While collaboration between in-service teachers and software design students appeared beneficial based on the initial needs assessment, precautionary measures were implemented. Before production began on the software development project, it was deemed necessary to assign tasks and specify roles. This was an attempt to prevent the needs of either party from being ignored, which could potentially foster animosity towards future collaboration and could hinder the completion of this particular project. As such, a collaborative model that takes into account such problems as role ambiguity and project coordination was developed specifically for this type of project (see Fig. 1). Descriptions of each phase of the model are detailed in the following section along with narrative describing the outcomes of each phase for this particular software development project.

Description

The Software Collaboration Model is based on traditional instructional design principles and is described in terms of phases (interest, equipment, etc.) but also considers the participants (software students, teacher). For each phase, participants have specific duties to perform or decisions to make. Just as a good instructional design model will lead to good instruction, when these phases are adhered to, it is postulated that the result will be a final product that benefits all parties involved. However, if at any time during the project the needs of one of the parties are not being met and a resolution does not occur, it is suggested that the collaboration be terminated. Without an end product that benefits both parties, the collaboration does not serve the intended purpose and the use of the model is nothing more than a formality.

Interest

The first phase of this model is critical to the successful completion of a product. The first phase is entitled "Interest" and involves two separate tasks. The first task is to identify the roles of the participants. According to traditional instructional design principles, it is crucial to identify the roles of those participating in the instructional design process (Dick & Carey, 1996; Kemp, Morrison, and Ross, 1994). Once roles are determined and agreed upon, there is little to no ambiguity as to the purpose of each participant throughout the process.

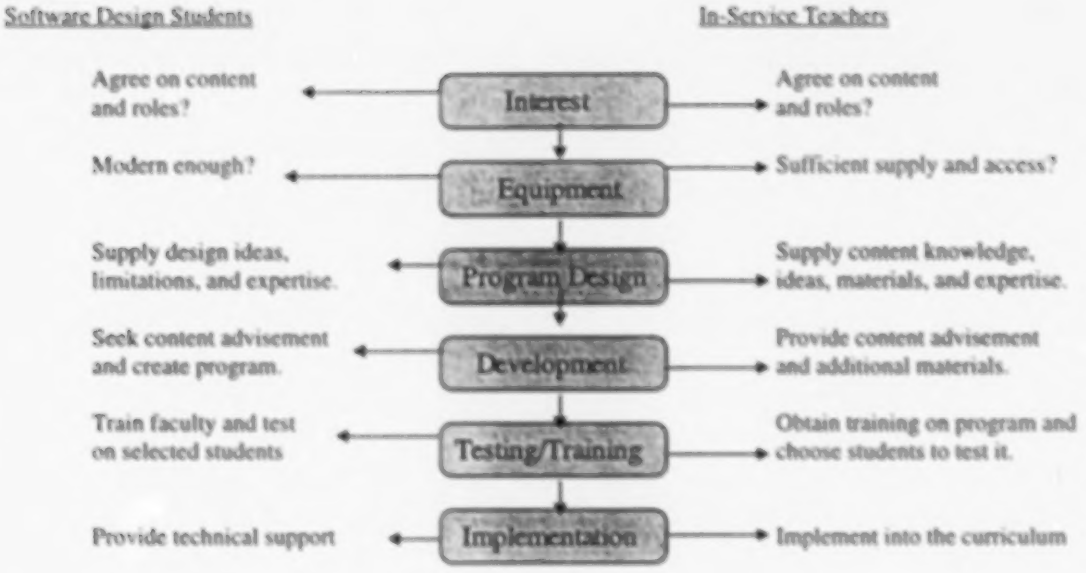

Figure 1. Software Collaboration Model

For this particular project, an introductory meeting was held to assign roles. The teacher was identified as the content expert and the software design students as the programming experts. It is no coincidence that the outcomes of this task mirrored the identification of subject-matter experts and media specialists, tasks identified in the instructional design models created by Kemp, Morrison, and Ross (1994) and Dick and Carey (1996).

The second task involved in the "Interest" phase includes the identification of the content to be used. Because this is a project that is to benefit both parties, both parties must be interested in developing the same content area. This step is very similar to the specification of content step in the Gerlach & Ely Model (Gerlach & Ely, 1980) and the identification of needs and goals step of the Dick and Carey Model (Dick & Carey, 1996).

Essentially, the purpose of this task is to identify the needs of both parties and specify the content area that will meet those needs. The content area that was specified in the introductory meeting for this specific project was 6th grade math. This content area was particularly important to the teacher, but was also an area in which the software students felt they could market their program.

Equipment

Because the instruction being considered is computer-based, the second phase of this collaborative model involves the identification of and access to sufficient equipment. Design models developed for computer-based instruction by Hannafin and Peck (1988) and Tripp and Bichelmeyer (1990) assert that the identification of computer resources is an essential aspect to the design process. As with those models, this particular model forces the participants to identify and recognize limitations that equipment may have on instructional development.

In this particular project, the design students toured the school and determined that the equipment in the school was modern enough to run the type of program that they needed to produce. At the same time, the teacher examined the computer lab schedule and determined that she had a sufficient supply of computers as well as adequate access. As a side note, had either of the participants felt that their needs were not being met due to equipment constraints or varying interests, the software project would have ceased at this point.

BEST COPY AVAILABLE

Program Design

As with the "Equipment" phase, the "Program Design" phase needs to consider the limitations of the media. However, other design limitations must also be considered at this point. In order to accomplish this, the subject-matter expert (teacher) and media experts (design students) must collaborate to determine the feasibility of several design options given the chosen content area. This phase is similar to the step in Dick and Carey's (1996) instructional design model where the selection of instructional materials and development of instructional strategies occurs. In essence, the specific focus of this phase is the mental creation of what the instruction will look like when finished, taking into consideration equipment, programming, and content limitations.

The specifics of this phase for this particular project are detailed and lengthy. Therefore, the report on this section is somewhat abbreviated. In this particular phase, the teacher became the center of attention as she supplied the content materials and expertise in the content area. She discussed several math units that could be covered and provided presentation ideas, instructional methods, and materials to the software design students. The students, along with the teacher, discussed the ideas while taking into consideration programming abilities and time limitations. In the end, specific units were determined and the teacher provided books and resources to the design students. Presentation and instructional methods were also determined for each unit. Both the teacher and the design students brainstormed potential methods and when they were finished, everyone had a concrete mental picture of what the instruction would look like when complete.

Development

While the subject-matter expert (teacher) is the center of attention in the previous phase, the media experts (design students) are the center of attention in the "Development" phase. During this period, the media experts are charged with the task of actually creating the computer-based instruction, while the subject matter expert (teacher) provides content advisement and any additional materials that are needed. This phase is in accord with the design phase described in the Rapid Prototyping Model (Tripp & Bichelmeyer, 1990) and the development phase of the Hannafin and Peck CAI Model (Hannafin & Peck, 1988). Ultimately, the outcome of this phase is a prototype program ready for testing.

During the "Development" phase of this particular project, several activities occurred that hindered the remaining project phases. First and foremost, the media experts (students) found that they were quite the optimists when designing the program and had not been realistic in some of their design decisions. Because of the linearity of this collaborative model, there were no opportunities to go back to previous phases and revise. Thus, the students felt as though they were constantly rushed during the production of the computer software and, as such, did not produce the level of quality that they had anticipated.

The second barrier that appeared during this phase was one involving communication. Up until this point, each phase was addressed in face-to-face meetings between the teacher and the design students. However, once development began, the meetings ceased to occur and communication was limited to e-mail. As students found that they needed more content resources than was initially provided by the teacher, they attempted to contact the teacher via e-mail. Because the teacher was not a regular user of e-mail and did not respond immediately to their requests, the students felt as though their progress was impeded due to the lack of communication with the teacher.

The aforementioned factors, when combined, led to a feeling of non-cooperation and frustration on the part of the media experts (students). As such, the students decided to fore-go the use of the subject-matter expert (teacher) and continue the development on their own. Hence, as was discussed at the beginning of this model, one party felt that their needs were not being met and, accordingly, the project as whole was dissolved.

Testing/Training

The "Testing/Training" phase of the Software Collaboration Model involves the testing of the program and the training of the teachers. This closely resembles the phase in the Rapid Prototyping Model (Tripp & Bichelmeyer, 1990) where the product is prototyped and researched. The task of the design student during this phase involves the training of faculty who plan to use the new software. The design students are also responsible for testing the program on selected students to help identify potential problems. Those students are chosen by the teacher, who is also responsible for receiving training on the new software at this time. Essentially, the outcome of this phase includes a trained teacher who can utilize the program and a prototype session in which potential problems are identified and fixed before the actual implementation. (Due to the fact that the design students in this particular project chose to dissolve their involvement, specific data for this phase is not available.)

Implementation

The final phase of the collaboration model involves the mass implementation of the program into the curriculum. The task of curriculum integration rest with the teacher, as does the scheduling of the computer lab. At this point, the primary task for the design students is to become a provider of technical support and assistance when needed. The design students may also take this opportunity to gather Beta data concerning their program. This phase most closely resembles the task in the Rapid Prototype Model (Tripp & Bichelmeyer, 1990) known as the "Install and Maintain System" step. (Due to the fact that the design students in this particular project chose to dissolve their involvement, specific data for this phase is not available.)

Revisions to the Model

Based on the results of the initial prototype project of the Software Collaborative Model, specific revisions have been recommended and implemented. First and foremost, the need for continual adaptation and revision of each phase is a necessity. Because the previous model lacked the revision process, the software design students felt undue pressure to adhere to content decisions that were made during the design phase even though they were found to be impractical during the development phase. To adjust and allow for revisions, the model has been updated and now includes a revision process at the design, development, testing/training, and implementation phases (see Fig. 2).

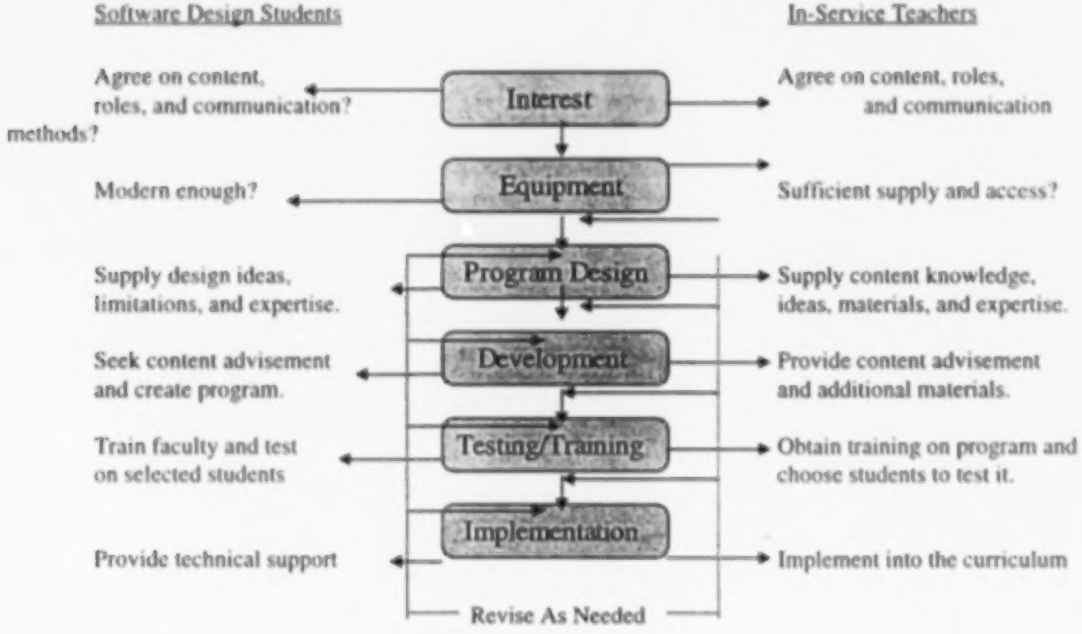
Software Design Students
In-Service Teachers
Agree on content, roles, and communication? methods?
Interest
Agree on content, roles, and communication
Modern enough?
Equipment
Sufficient supply and access?
Supply design ideas, limitations, and expertise.
Program Design
Supply content knowledge, ideas, materials, and expertise.
Seek content advisement and create program.
Development
Provide content advisement and additional materials.
Train faculty and test on selected students
Testing/Training
Obtain training on program and choose students to test it.
Provide technical support
Implementation
Implement into the curriculum
Revise As Needed

Figure 2. Revised Software Collaboration Model

The second revision to the Software Collaboration Model involves communication. Rather than address methods of communication in the initial "Interest" phase, the topic was overlooked and eventually led to the disintegration of the prototype software design project. To prevent such an instance from occurring again, a task has been added to the "Interest" phase (see Fig. 2). Initially, the two tasks of the interest phase included agreeing on content and roles. The revised model adds the task of predetermining communication methods to this phase. By predetermining the methods to be used, participants utilizing the model may encounter less difficulties with the communication process.

The third and final revision that is undergoing consideration involves the instructional design component of the model. During the "Program Design" phase, the present model relies on the knowledge of both the teacher and the software design students to ensure that appropriate instructional design procedures are followed. However, it is proposed that the involvement of a third party who specializes in instructional design may enhance the instructional quality of the final produced program. At the present, the current author is experimenting with this idea by adding a

BEST COPY AVAILABLE

masters level instructional technology student to the project. The benefits for both the instructional technology student and the final software program are currently being explored, but have yet to be added to the model. If successful, this aspect will be added to the Software Collaboration Model.

Conclusion

This paper has briefly described the needs and benefits of a collaboration between senior software design students and public teachers. Based on those needs and benefits, the Software Collaboration Model, which mirrors aspects of traditional instructional models, was developed. To test the collaborative model, a group of software design students and a teacher were chosen to participate in the collaborative development of a software design project. The results of this prototype project that attempted to utilize the Software Collaboration Model was reported. From those results, the areas of communication, revision, and instructional design were identified as potential areas in need. A revised version of the model was developed and described that took into account two of these three areas. At this point, the call is for future prototyping and, if necessary, future revision to the model.

References

Braswell, R. (1994). The top five reasons not to use multimedia. In Eds. W. Michael Reed, John K. Burton & Min Liu, Multimedia and Megachange. New York, The Haworth Press, Inc., 281-284.

Dick, W. & Carey, L. (1996). The systematic design of instruction. Glenview, IL: Scott, Foresman, and Company.

Gerlach, V. S., & Ely, D. P. (1980). Teaching and media: A systematic approach (2nd ed). Englewood Cliffs, NJ: Prentice Hall.

Golas, K. C. (1993). Estimating Time to Develop Interactive Courseware in the 1990s. Proceedings of the 15th Interservice/Industry Training Systems and Education Conference, Orlando, Florida, November 29-December 2.

Hannafin, M. J., & Peck, K. L. (1988). The design, development, and evaluation of instructional software. New York: Macmillan Publishing Company.

Kemp, J. E., Morrison, G. R., & Ross, S. M. (1994). Designing effective instruction. New York: Maxwell MacMillan International.

Maddux, C. D. (1994). Multimedia, hypermedia, and the culture of schooling. In Eds. W. Michael Reed, John K. Burton & Min Liu, Multimedia and Megachange. New York, The Haworth Press, Inc., 21-26.

Oughton, J. M., & Krawchuk, C. A. (1997). Debating the uses of hypermedia in K-12 schools. Presented at the Annual Conference for the Eastern Educational Research Association, Hilton Head, SC.

Tripp, S., & Bichelmeyer, B. (1990). Rapid prototyping: An alternative instructional design strategy. Educational Technology Research and Development, 38(1), 31-44.